|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 6**

|  |  |
| --- | --- |
| **Реализация и исследование алгоритма построчного затравочного заполнения сплошных областей.**  **Студент** Жигалкин Д.Р  **Группа** ИУ7-45Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_**Куров А. В.**\_\_\_\_\_** |  |

Москва.

2020 г.

**Цель работы:** Реализация и исследование алгоритма построчного затравочного заполнения.

**Задание:**

Необходимо обеспечить ввод произвольной многоугольной области, содержащей произвольное количество отверстий. Ввод (вершин многоугольника) производить с помощью мыши, при этом для удобства пользователя должны отображаться ребра, соединяющие вводимые вершины. Предусмотреть ввод горизонтальных и вертикальных ребер. Должен быть предусмотрен ввод затравочной точки.

Пользователь должен иметь возможность задания цвета заполнения.

Работа программы должна предусматривать два режима – с задержкой и без задержки.

Режим с задержкой должен позволить проследить выполняемую последовательность действий.

(Задержку целесообразно выполнять после обработки очередной строки).

Обеспечить замер времени выполнения алгоритма (без задержки, с выводом на экран только окончательного результата).

Продемонстрировать возможность заполнения с помощью затравочного алгоритма произвольной области, ограниченной замкнутой кривой линией.

**Теоретический материал.**

Алгоритмы заполнения с затравкой используют для заполнения области затравочный пиксель, он расположен внутри заполняемой области.

Работа алгоритма состоит из 4 этапов.

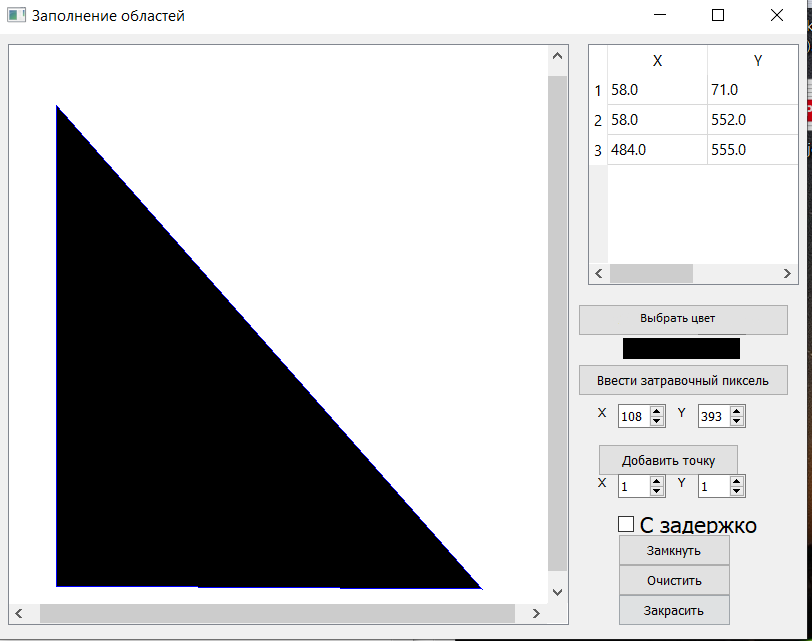
Затравочный пиксель извлекается из стека (pop), содержащий затравочные пиксели.

Интервал с затравочным пикселем заполняется влево и вправо от затравки вдоль сканирующей строки до тех пор, пока не будет найдена граница. В **xl** и **xr** запоминаются крайний левый и крайний правый пиксели интервала.

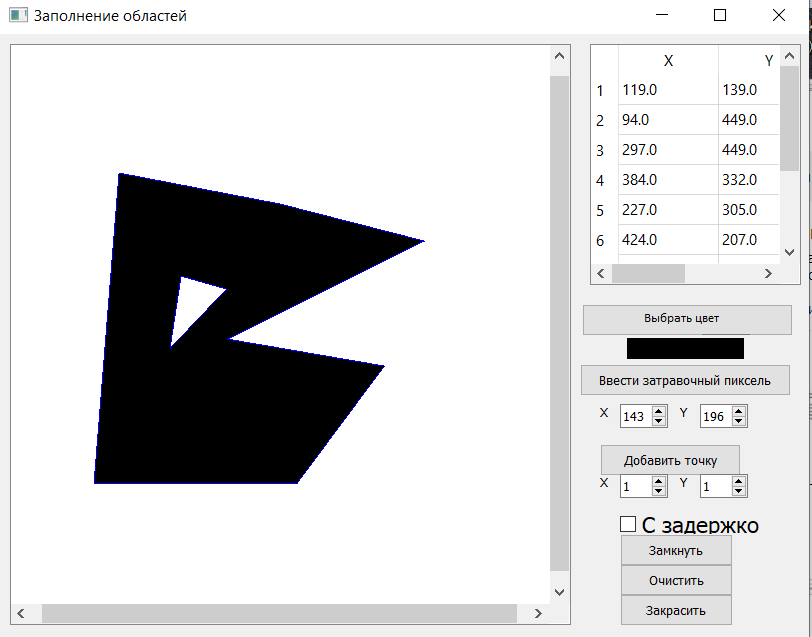
В диапазоне **xl** <= x <= **xr** проверяются строки, расположенные непосредственно над и под текущей строкой. Определяется, есть ли на них ещё не заполненные пиксели. Если такие есть, то в указанном диапазоне крайний правый пиксель в каждом интервале отмечается как затравочный и помещается в стек.

При инициализации алгоритма в стек помещается затравка(push), работа завершается при опустошении стека.

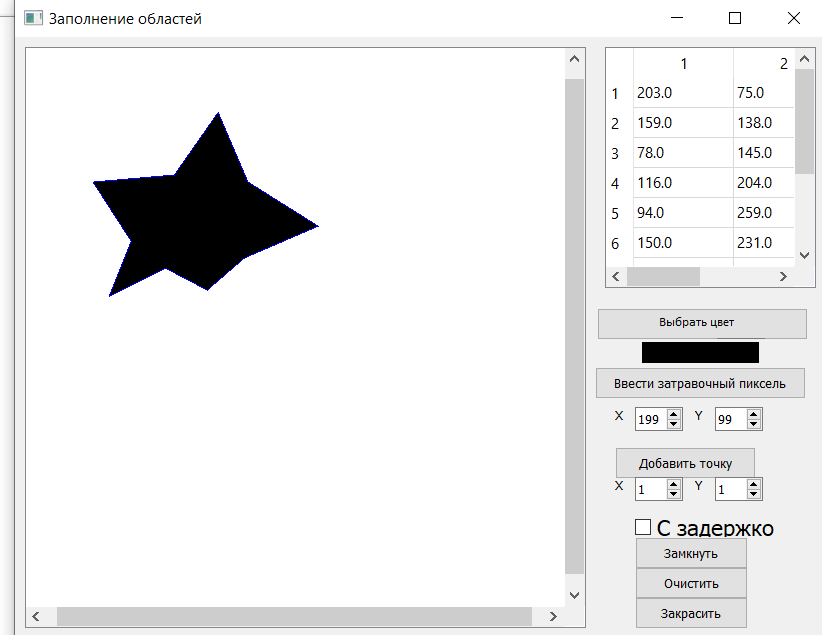
**Примеры работы программы.**



Time: 950 ms.



Time 1001 ms.

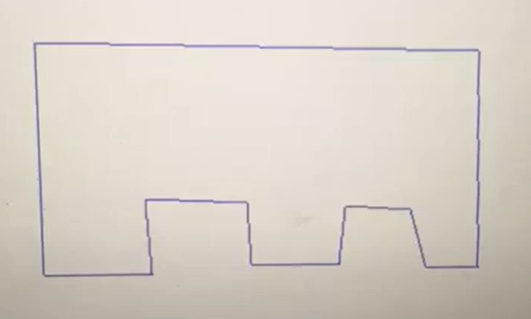


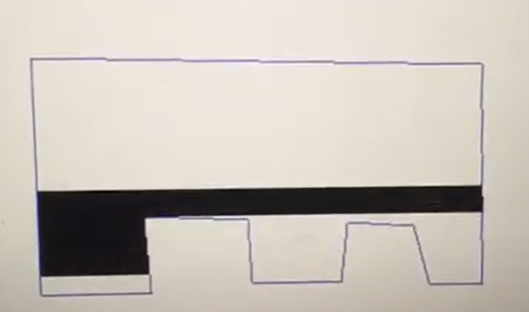
Время выполнения 200 ms.

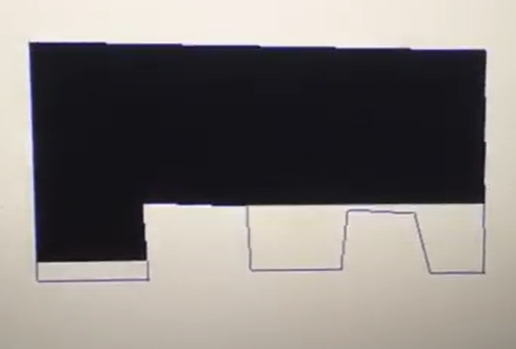
Работа алгоритма с задержкой (видео):

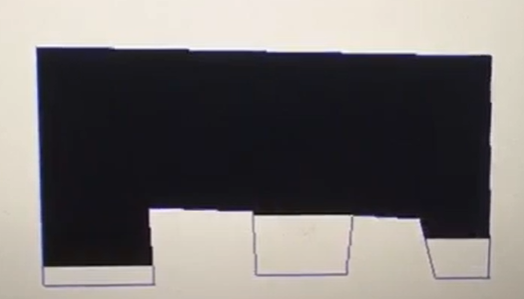
Ссылка на видео: <https://www.youtube.com/watch?v=ZXfWrvOuaqM>

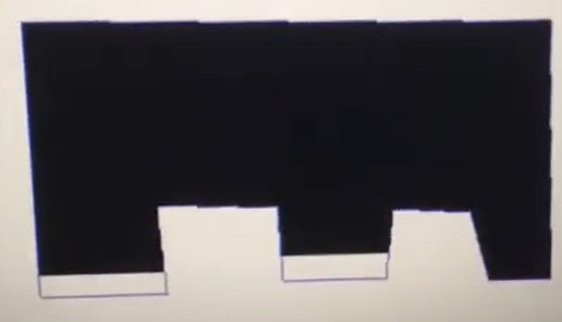
Кадры из видео:

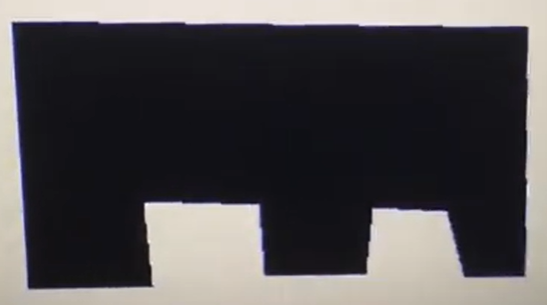












**Исходный код.**

